

### **AMENDMENTS TO THE SPECIFICATION**

**In paragraphs [0005], [0012], [0019] and [0028] please amend as reflected in the following marked-up versions of the paragraphs:**

**Please replace paragraph [005] with the following amended paragraph:**

The current art of asynchronous message-oriented communication provides for guaranteed messaging. Based on the current art, a messaging system such as IBM's MQSERIES ~~MQ Series~~ will guarantee the once and only once processing of a message. With the advent of Web Services and standards-based, service-oriented communication, the same concept has been extended to the delivery and processing of the input data of a service upon consumption.

**Please replace paragraph [0012] with the following amended paragraph:**

To achieve the foregoing objectives, ~~and~~ in accordance with the purpose of the invention as broadly described herein, ~~and with reference to Fig. 5,~~ the present invention provides methods, frameworks, and systems for providing a method for guaranteed invocation of optionally nested composite services. In preferred embodiments, this technique comprises: 1) associating the attribute required for the guaranteed invocation to the service interface definition (501); 2) providing the ability to overwrite those attributes for a service contained within another composite service, in the context of the containing service (502); 3) defining a wrapper interface with methods accommodating the guaranteed invocation of non-composite services (503); 4) creating a directed execution graph based on the definition of a composite service having one such graph per each embedded composite service (504); 5) creating a possibly nested Invocation Map corresponding to the state of execution of each instance of a composite service (505); 6) coupling a persistent context mechanism with each Invocation Map (506); 7) associating unique ~~ids~~ IDs to each service invocation while breaking the invocation into steps based on the directed execution graph (507); 8) recording each step (508); 9) performing each step of invocation and marking the results through the persistent context mechanism where a result of the invocation of a contained service is recorded (509); 10) upon a system failure, or inability to invoke a service due to a system failure such as connection failure, attempting to retry the invocation of the service from where it previously left the invocation, based on the state of the associated context mechanism (510); 11) upon the unknown state of the invocation of a particular service, invoking the service with the same ~~id~~ ID with which it was invoked prior to the unknown state (511).

**Please replace paragraph [0019] with the following amended paragraph:**

Assuming a visual software service composition and assembly tool, such as HyperService™ Studio, with a snapshot of a definition in FIGURE 2, and an automatic flow platform, such as the HyperService™ platform, ~~available at [www.nextaxiom.com](http://www.nextaxiom.com) provided by~~ NextAxiom® Technology Inc., we will now describe how such a composition tool and platform can be extended to accommodate declarative, automated guaranteed invocation for optionally, nested composite software services.

**Please replace paragraph [0028] with the following amended paragraph:**

Methods, systems, frameworks and software products are disclosed for semantic-based definition, and guaranteed invocation/consumption of nested composite software services containing other ~~composite~~ composite or atomic software services, such as but not limited to Web services.

**Please add the following new paragraph after the paragraph [0017]:**

FIGURE 5 depicts a flowchart of one embodiment of a method for guaranteeing invocation of composite software services.